

## Görev-12

### Amaç:

Bu görevde Görev-10'da ve Görev-11'de oluşturduğunuz controller, action, view ve entity'ler üzerine eklemeler yaparak devam edeceksiniz. Görev-12 ve Görev-13 tamamlandığında action ve view'lar ile veritabanı işlemlerini birbirine bağlayarak CRUD işlemlerinin gerçekleştirilmesi amaçlanmıştır. Ayrıca kullanıcıdan alınan verilerin validasyonlardan geçirilerek kısıtlamalara uygunluğu kontrol edilecektir. Görev-12'de controller ve action'ların bir bölümü, Görev-13'te ise geri kalan kısımlar benzer şekilde tamamlanacaktır. Aşağıda Görev-12'de tamamlanması beklenen kısımlar belirtilmiştir;

### e-Ticaret MVC

Açıklama	Controller	Action
Anasayfa	Home	Index
Hakkımızda	Home	AboutUs
İletişim	Home	Contact
Ürün Listeleme	Home	Listing
Ürün Görüntüleme	Home	ProductDetail
Ürün Oluşturma	Product	Create
Ürün Düzenleme	Product	Edit
Ürün Silme	Product	Delete
Ürüne Yorum ve Yıldız Yapma	Product	Comment

### Admin MVC

Açıklama	Controller	Action
Anasayfa	Home	Index
Kategori Oluşturma	Category	Create
Kategori Düzenleme	Category	Edit
Kategori Silme	Category	Delete
Ürünü Silme	Product	Delete

## Yapılacaklar:

- DbContext'i ilgili Controller'ların constructor'larına parametre olarak ekleyin
- CRUD işlemi olan action'larda kullanıcıdan veri alınan yerlerde “{ControllerAdı}{ActionAdı}Viewmodel” formatına benzer şekildeki view model'ler aracılığıyla verinin alınmasını sağlayın (*model binding*). View modelleri proje içerisinde ayrı bir klasörde oluşturun (Models veya ViewModels klasörü olabilir)
- View modellerdeki property'lere, amacına uygun ve ilgili entity'nin validasyon kurallarıyla uyumlu validasyon attribute'larını ekleyin,
- View'larda validasyon sonuçlarının görüntülenebilmesi için ilgili “asp-validation-summary” kısımlarını ekleyin,
- Client side ve server side validasyon işlemleri için gerekli kodları ekleyin,
- İlgili action'larda, View model'lerden gelen bilgilerle gerekiyorsa entity ya da entity'ler oluşturup bu entity'ler ve DbContext aracılığıyla veritabanı işlemlerini gerçekleştirin
- View'a ilgili işlem sonucunun sadece mesaj olarak döneceği durumlarda ViewBag veya ViewData aracılığı ile ilgili mesajı ekleyip View'da görünmesini sağlayın (bootstrap'deki alert componenti bu iş için uygun olabilir)
- CRUD işlemi sonrası entity'nin son halinin View'da görünmesi gereken durumlarda ise sadece View'da görünecek alanları içeren bir ViewModel oluşturup bu model aracılığıyla View'da verilenin görünmesini sağlayın

## İpuçları:

- View'larda, “@model” keyword'ü ile doğru ViewModel'in kullanılmasını sağlayabilirsiniz,
- View'lardaki form ve input elementlerinde tag helper'lar aracılığı ile model binding ve data validasyon işlemlerinin otomatik ve sorunsuz bir şekilde gerçekleştirilebilmesini sağlayabilirsiniz,
- Client-side validasyon için, yeni bir MVC projesi oluşturulduğunda otomatik olarak oluşan “Views>Shared” klasörü altındaki “\_ValidationScriptsPartial.cshtml” dosyasını kullanmanız işinizi kolaylaştırabilir

**Teslim:** Bu görevde teslim edeceğiniz proje, sadece Görev-12'de yapılanları değil; Görev-10, Görev-11 ve Görev-12'deki değişikliklerin *tamamını* içermelidir, lütfen dikkat ediniz. Dosyalarını içeren klasörü zip'leyip classroom'da Görev-12 paylaşımına ek olarak ekleyin.